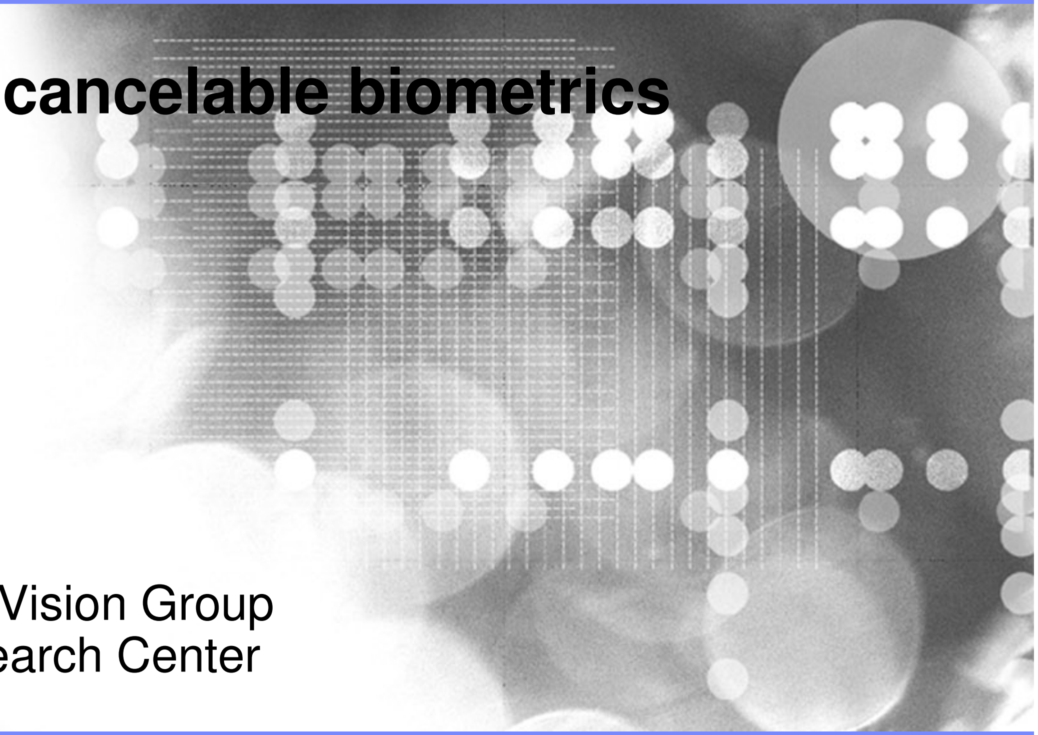
IBM Research

Performance evaluation of cancelable biometrics



Nalini K Ratha*
Exploratory Computer Vision Group
IBM T. J. Watson Research Center
Hawthorne, NY 10532

***Joint work with members of biometrics research team**

Cancelable Biometrics

- Intentional repeatable distortion
  - Generates a similar signal each time for the same user
- Compromised scenario:
  - a new distortion creates a new biometrics
- Comparison scenario:
  - different distortions for different accounts
- Backwards compatibility
  - Representation is not changed.

Cancelability requirements of the transform

1. The intrinsic strength (individuality) of the biometric should not be reduced after transformation. (Constraint on FAR)

$$D(x_1, x_2) > t \Rightarrow D(T(x_1), T(x_2)) > t$$

2. The transformation should be tolerant to intra-user variation (Constraint on FRR)

$$D(x_1, x_2) < t \Rightarrow D(T(x_1), T(x_2)) < t$$

3. The original should not match with the transform,

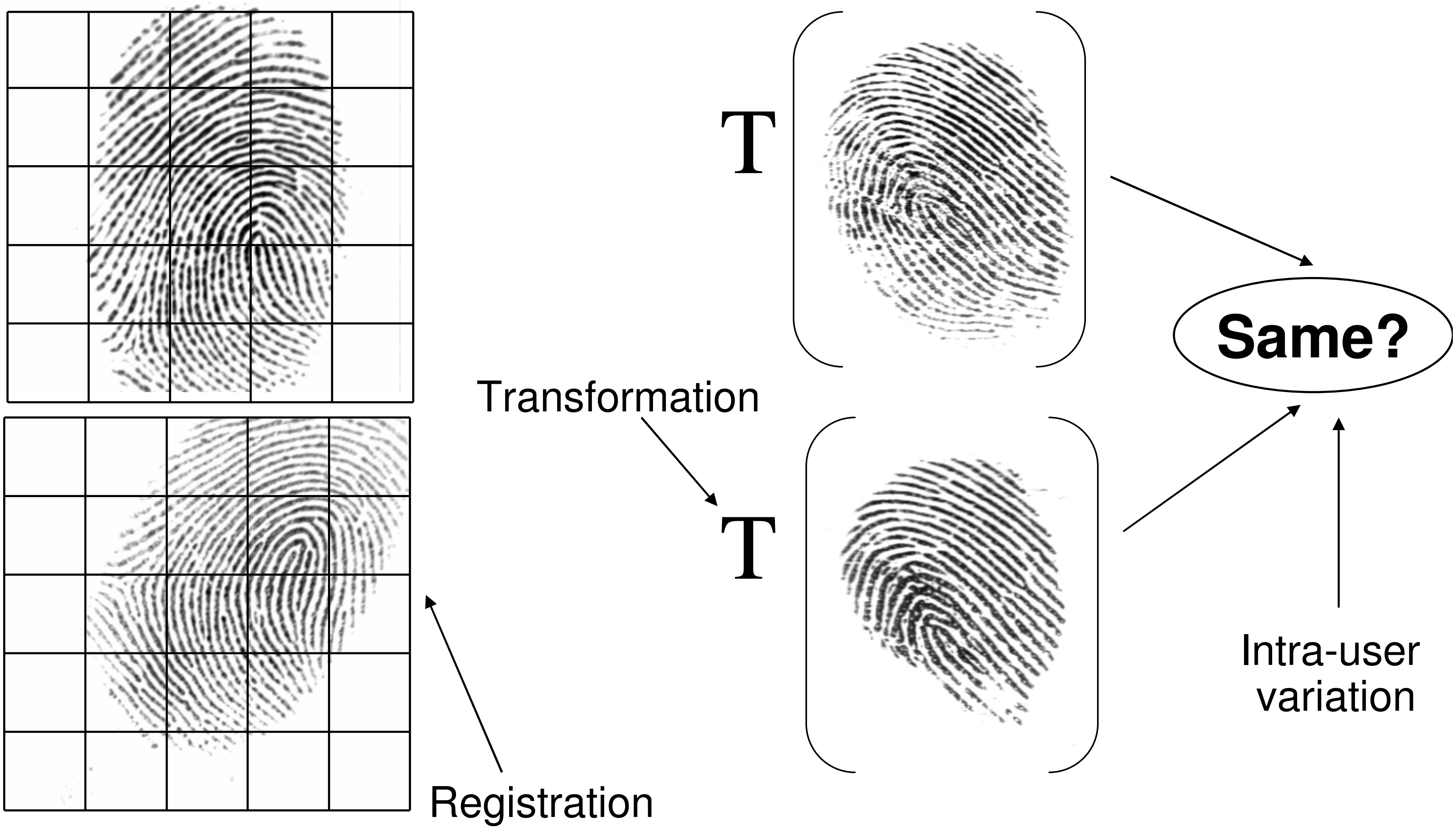
$$D(x, T(x)) > t$$

4. Different transforms of the same user should not match with each other

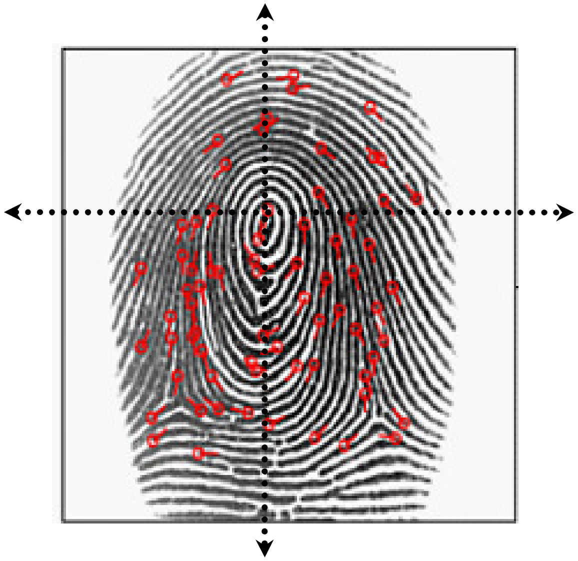
$$D(T_1(x), T_2(x)) > t$$

Registration based

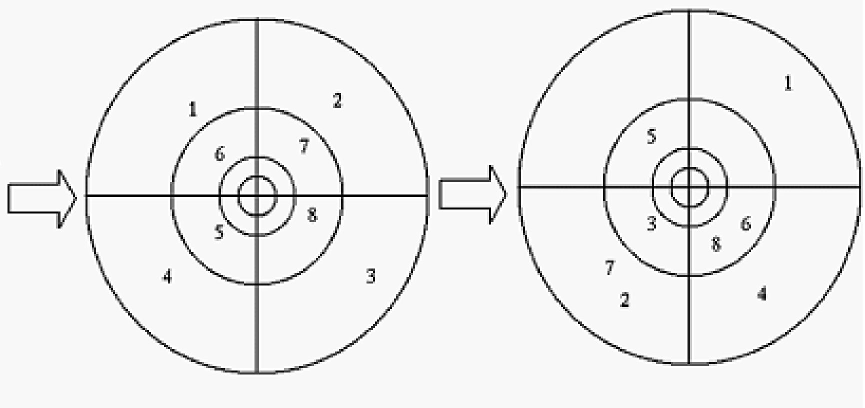
Challenges


T
Transformation
T
Same?
Intra-user variation
Registration

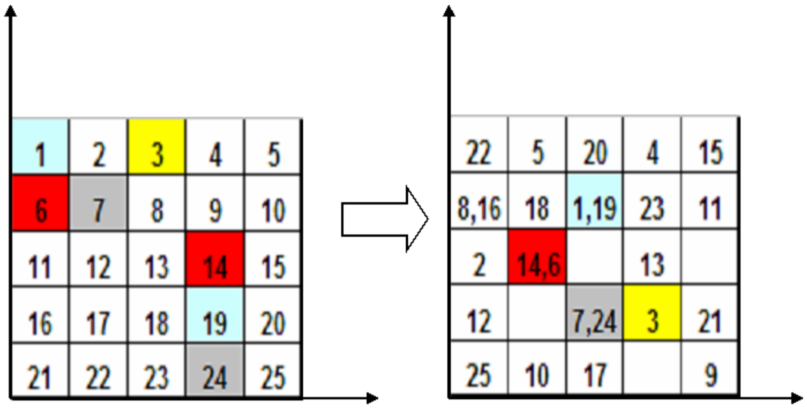
Feature Domain Transformation



Feature Extraction



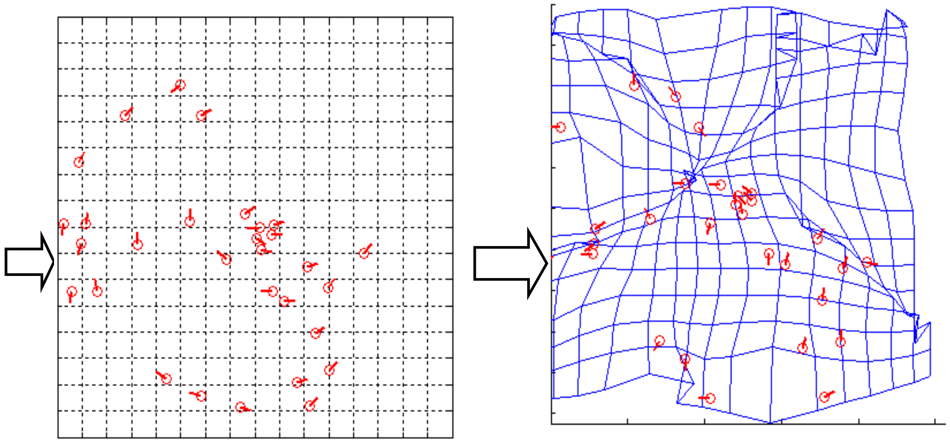
Polar Transformation



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

22	5	20	4	15
8,16	18	1,19	23	11
2	14,6		13	
12		7,24	3	21
25	10	17		9

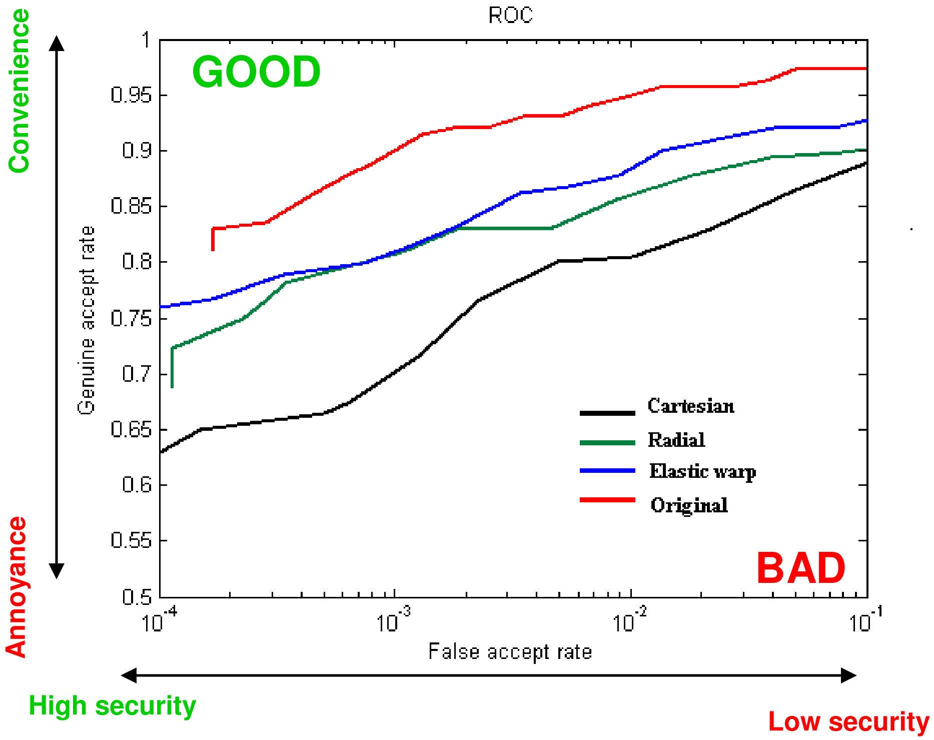
Cartesian Transformation



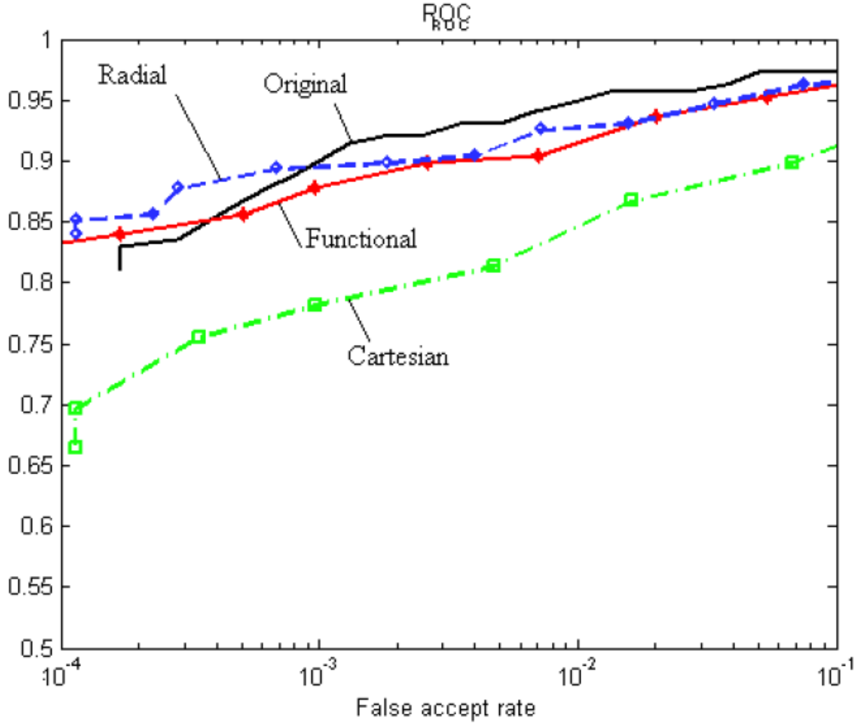
Surface Folding Transformation

How does it affect accuracy?

Same transform for all users



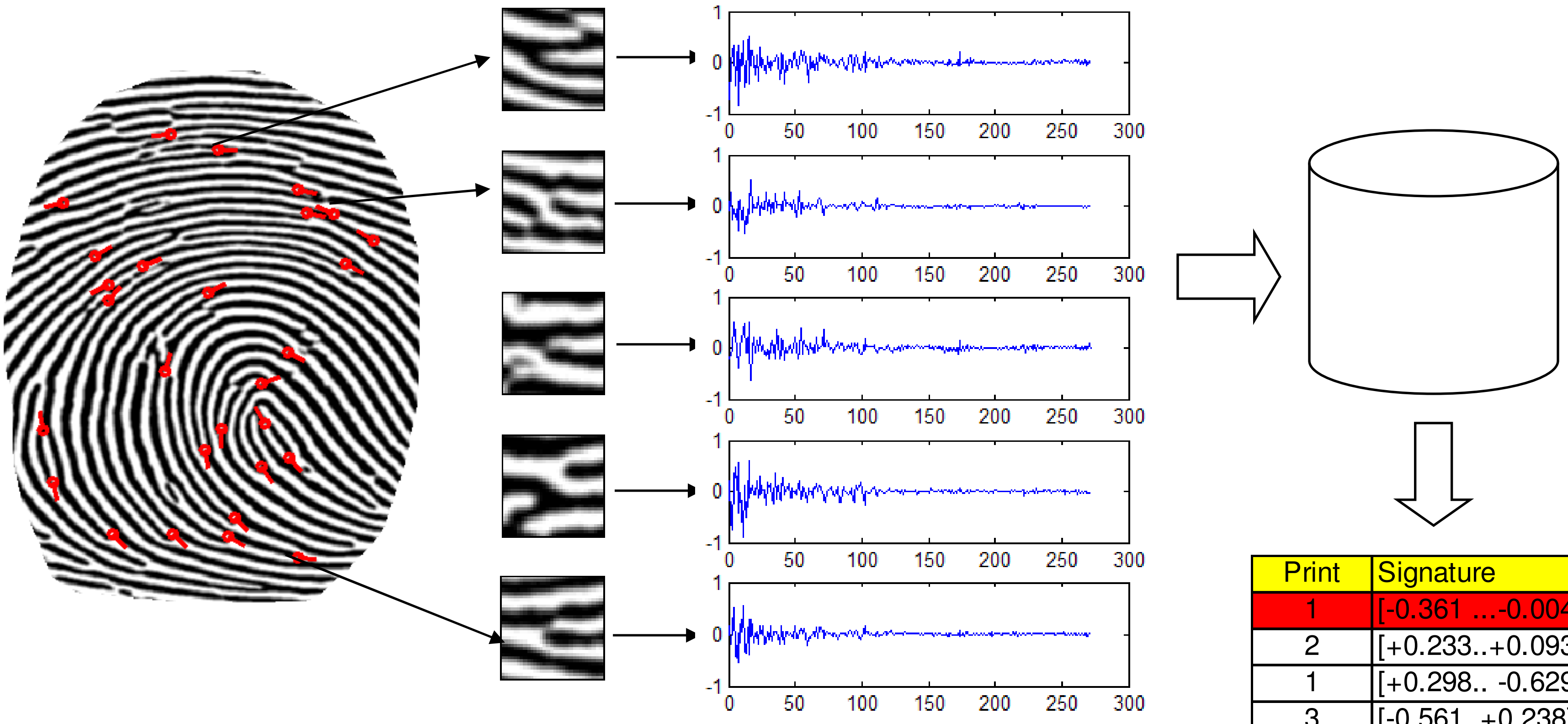
Different transforms for different users



- Results reported in
  - “Cancelable biometrics: A case study in Fingerprints”, ICPR 06
  - “Generating cancelable fingerprint templates”,IEEE PAMI

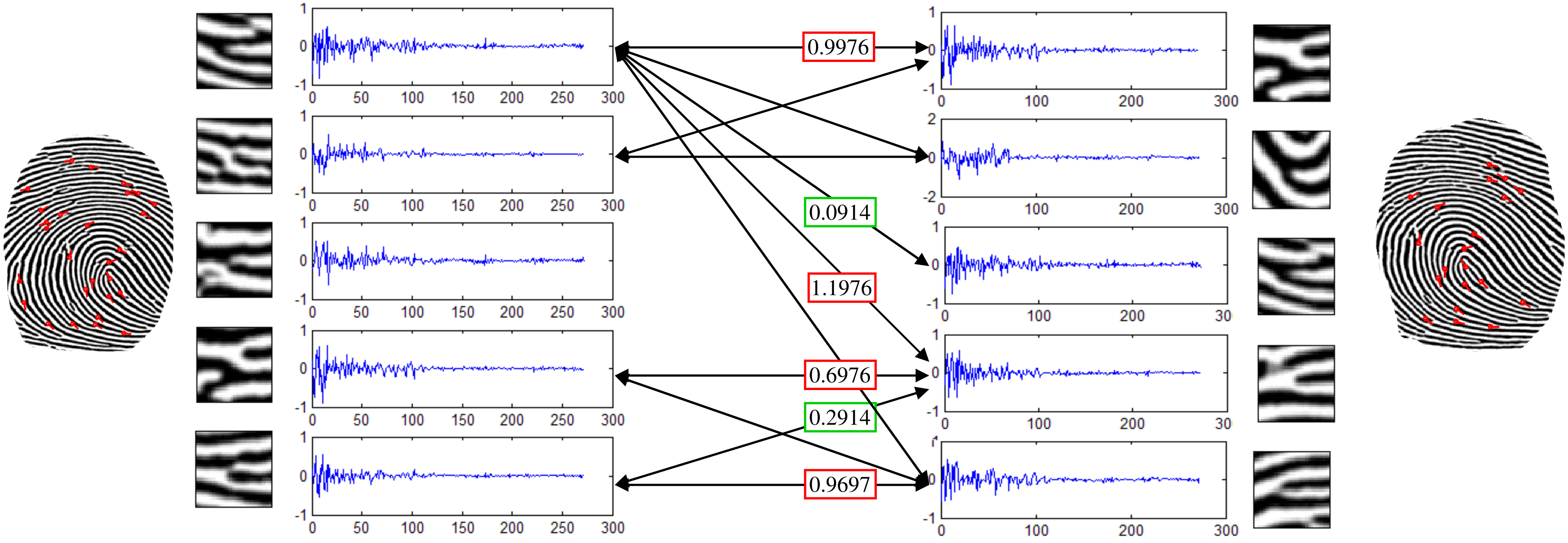
Registration free

Enrollment



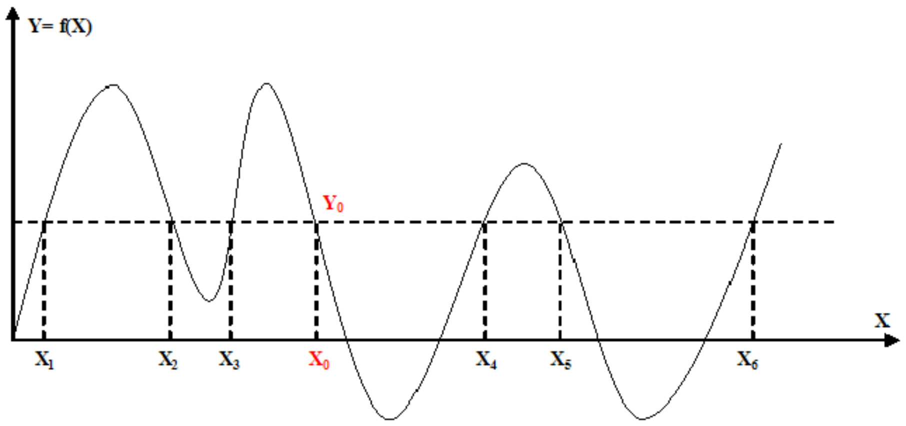
Print	Signature
1	[-0.361 ...-0.004]
2	[+0.233..+0.093]
1	[+0.298.. -0.629]
3	[-0.561..+0.238]
2	[+0.321..-0.563]
1	[-0.476..+0.672]
1	[+0.786..-0.054]
1	[-0.189..+0.034]

Verification



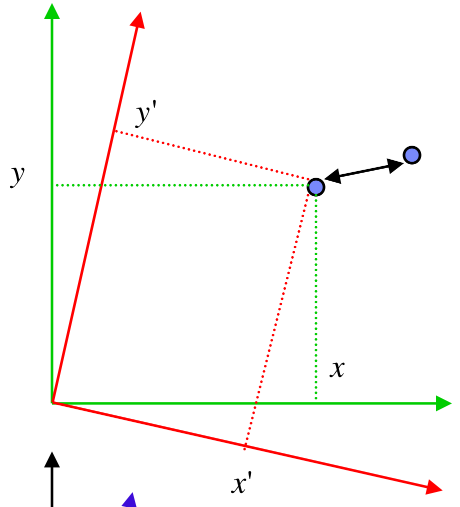
Cancelable methods

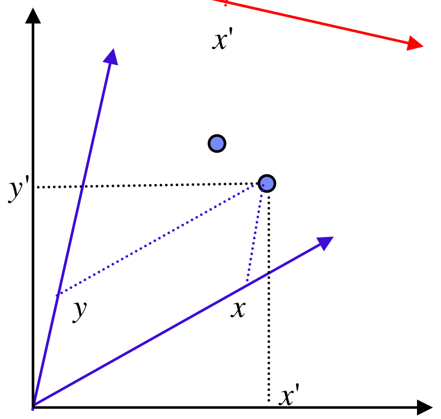
- Can we avoid storing the original patch signatures?
- Ways to transform/hide the feature vector
  - Encryption - representation too unstable for encryption
  - Polynomial transformation
  - Random projection- fits well with NDP distance



Polynomial transformation

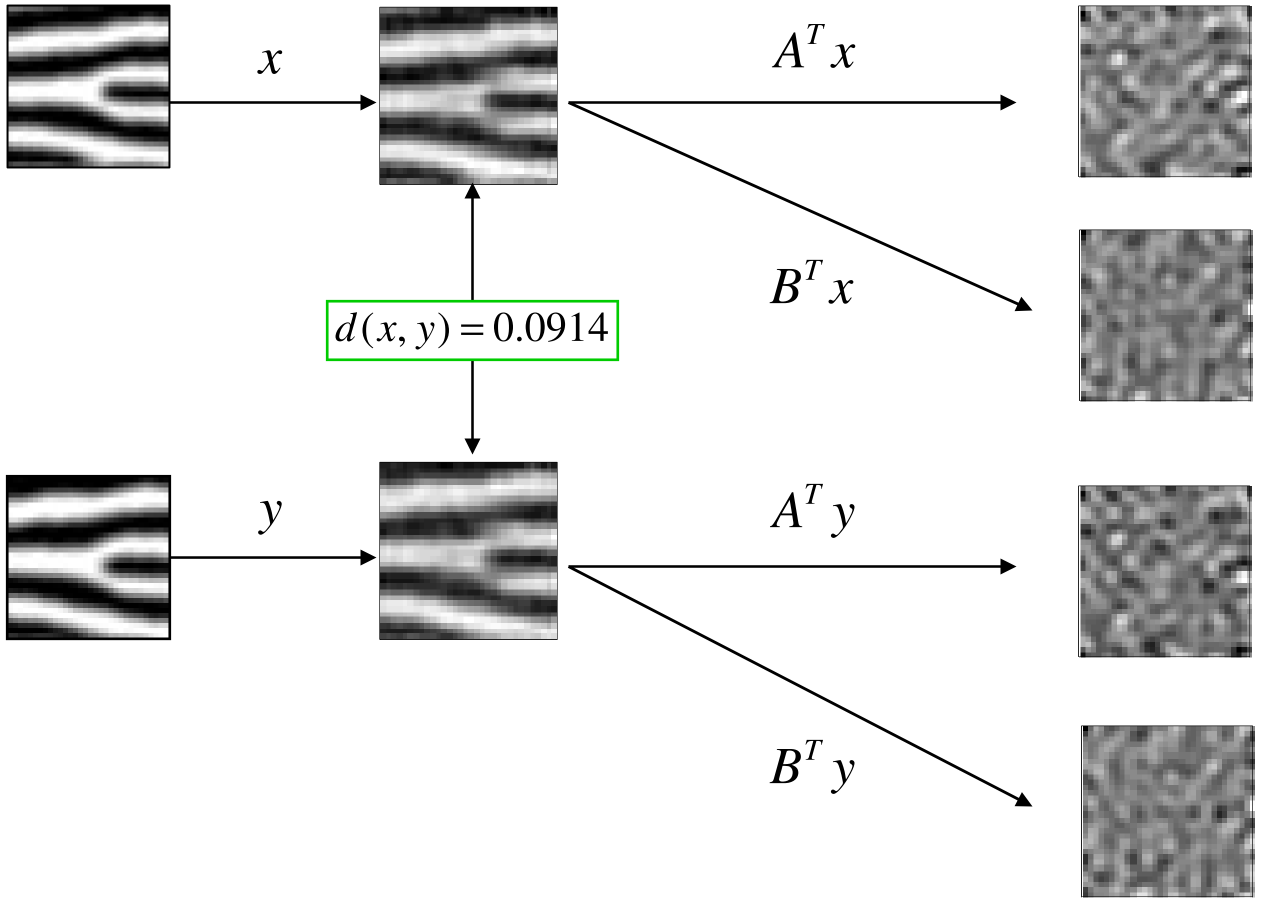
Preferred: Ortho normal projections





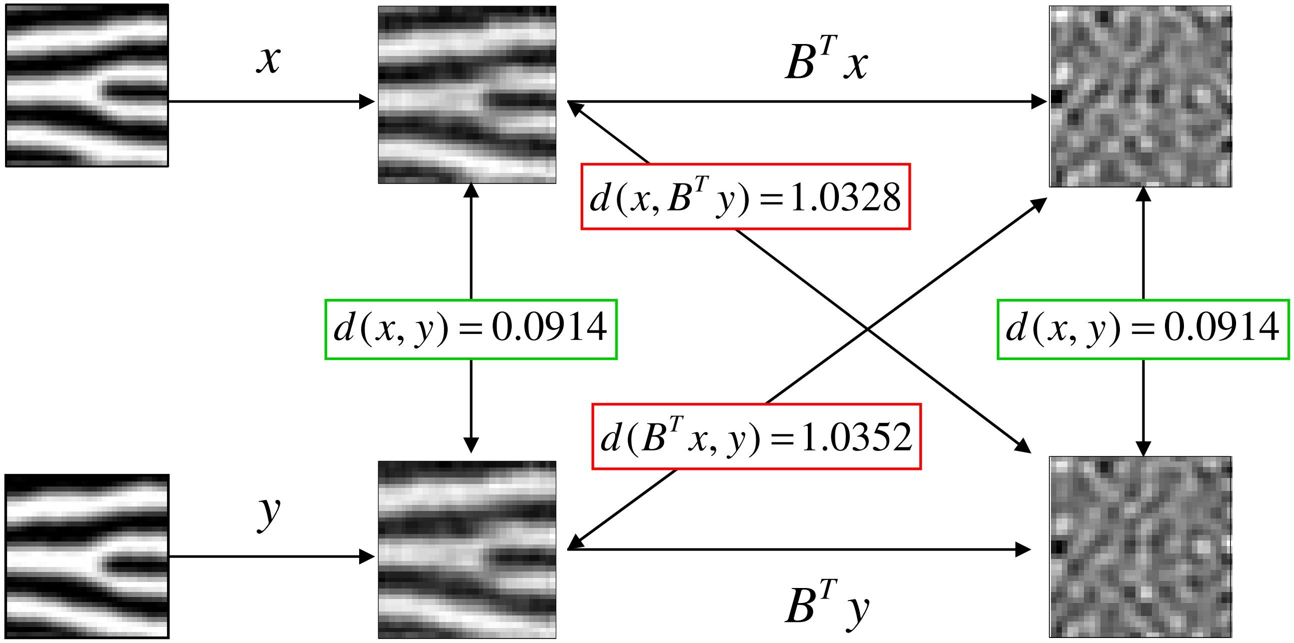
Random Projections

Cancelability (2)



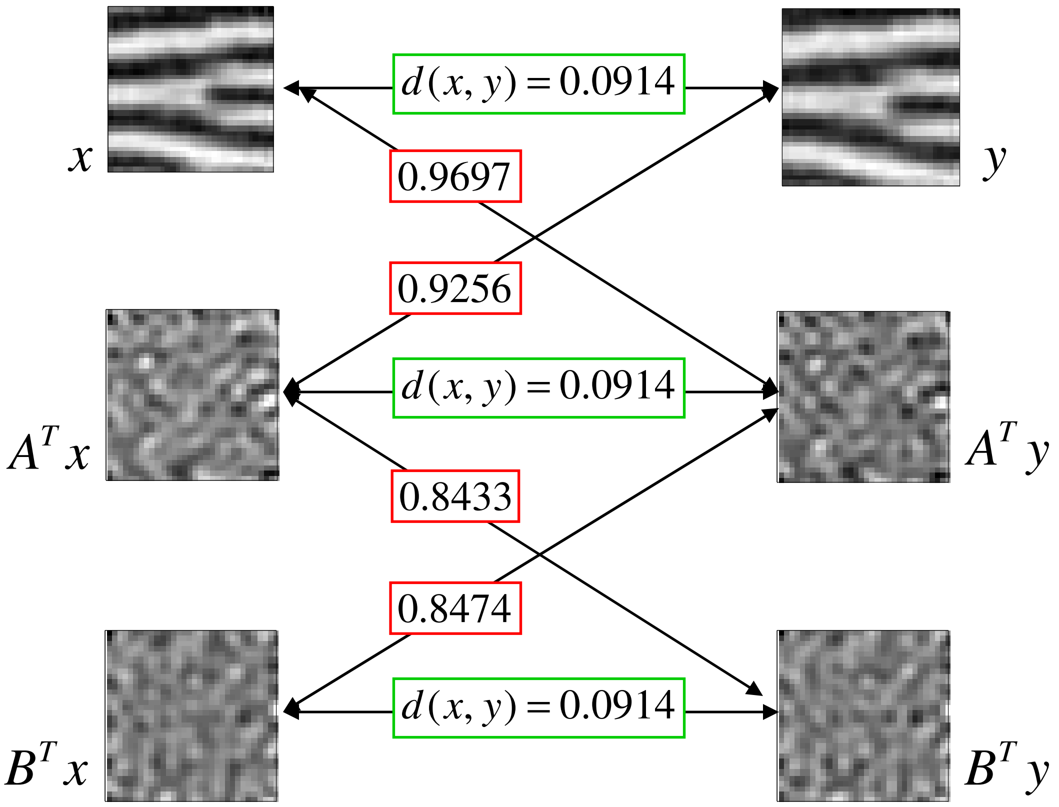
- Each patch can be used to produce multiple transforms

→Cancelability (3)



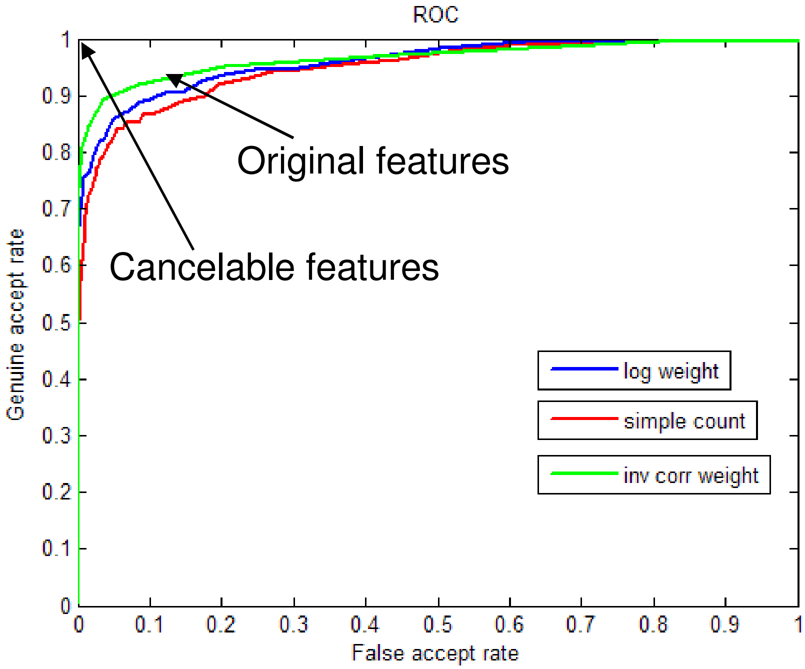
- Original match among themselves
- Transforms match among themselves
- Transform does not match with original

Cancelability (4)



- Score more than 0.5 is a mismatch
- Different Transforms don't match with each other

Empirical Results (1)



- Patch based verification
  - Performance is less than geometry based matchers (62% GAR at 0.01% FAR)
- Cancelabilility
  - Complete separation (100% GAR,0% FAR) achieved by having separate transforms for separate individuals
- Diversity of key space
  - Complete separation (100% GAR,0% FAR) achieved for separate (188) transforms of the same individual.
- Non invertiblity
  - Complete separation (100% GAR,0% FAR) achieved for non-invertible construction as well

➢ **Perfect performance because uses entropy from key also**

➢ **If everyone uses the same key performance will not change because distances are preserved**

Increasing security: Two factor transformation

- The current construction is invertible

If we have the projecting matrix B, and the transform $\mathrm{T}(x) = B^T x$

$x = BT(x) = BB^T x$, can be recovered

- Can we increase security?

- Two factor transformation
  - The projection matrix B is constructed using two orthonormal matrices U,V

$$B = UV^T$$

$$UU^T = U^T U = VV^T = V^T V = I$$

$$BB^T = \left(UV^T\right)VU^T = U\left(V^T V\right)U^T = I$$

U, V are obtained by performing SVD on a random matrix $R = USV^T$

S is not recorded anywhere in the system.

U, V do not leak information about each other

- U and V can be separately stored separately (e.g. split between user and application?)
- Symmetric key, public key comparison

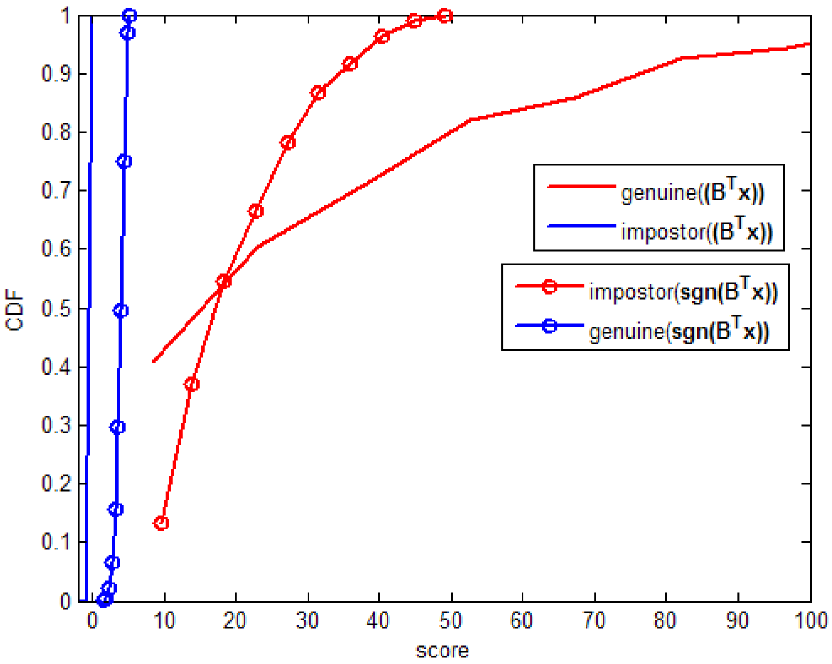
More security: Non-invertibility

- We can make the construction non-invertible by introducing some non-linearity

Define,

$$T(x) = \begin{cases} 1 & \text{if } B^T x > 0, (B = UV^T) \\ 0 & \text{otherwise} \end{cases}$$

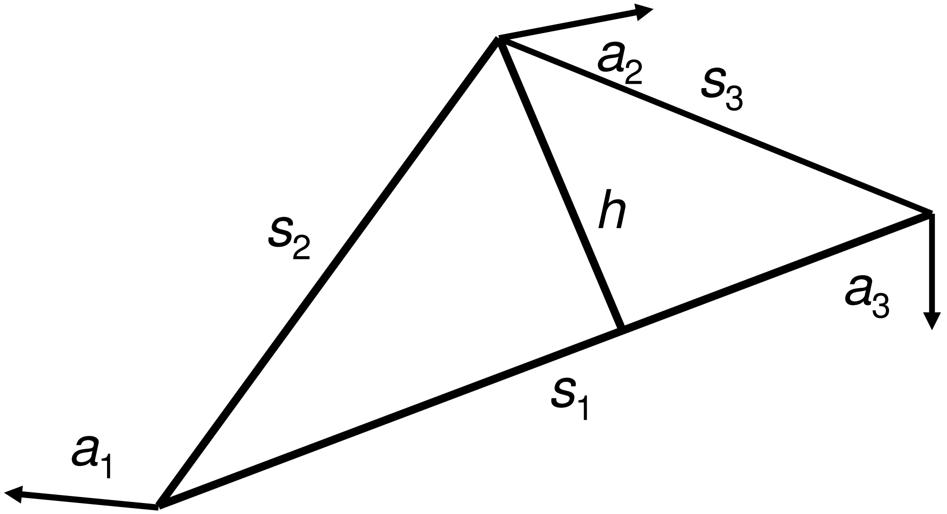
- Thus, even if U, V, T(x) are known, it is impossible to recover x from T(x)
- Advantages:
  - The construction is non-invertible
- Disadvantages
  - Brute force attack is easier. (More pre-images of $B^T x$ produce the same sign)



Score distributions for invertible and non-invertible construction

Invariant features

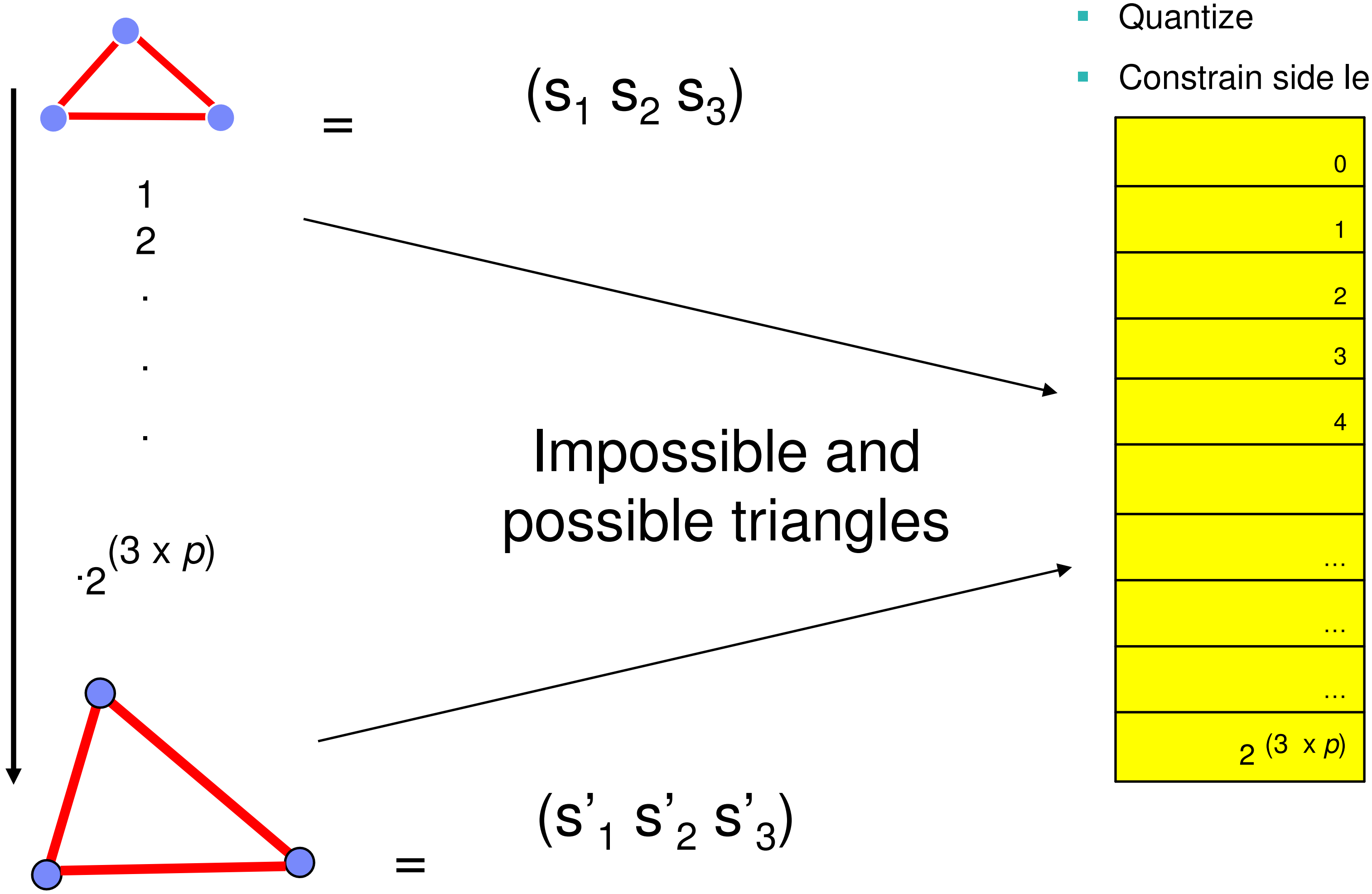
- Independent triangle features
  - The sides
- Dependent triangle feature
  - Height at largest side
- Fingerprint features
  - Minutiae angles with respect to triangle



$$\left.\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ s_1 \\ . \\ . \\ . \\ . \end{pmatrix}\right\} \text{INDEX}$$

Triangles can be enumerated

- Quantize
- Constrain side lengths



= $(s_1\ s_2\ s_3)$

1

2

.

.

.

$\cdot 2^{(3 \times p)}$

Impossible and possible triangles

0

1

2

3

4

…

…

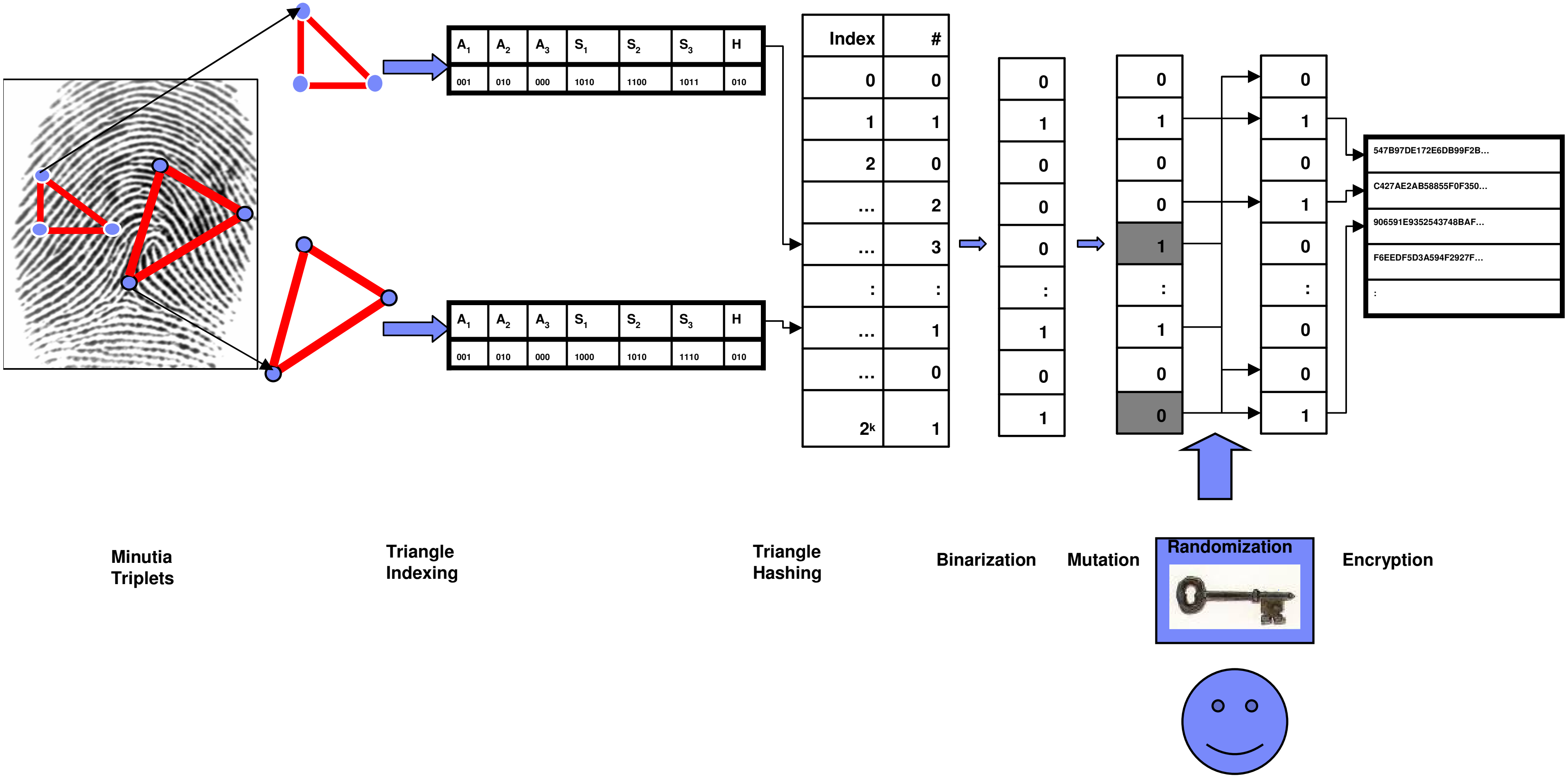
…

$2^{(3\ \times p)}$

= $(s'_1\ s'_2\ s'_3)$

s_1, s_2, s_3 quantized using *p* bits

Enrolment



A_1	A_2	A_3	S_1	S_2	S_3	H
001	010	000	1010	1100	1011	010

A_1	A_2	A_3	S_1	S_2	S_3	H
001	010	000	1000	1010	1110	010

Index	#
0	0
1	1
2	0
…	2
…	3
:	:
…	1
…	0
2^k	1

0
1
0
0
0
:
1
0
1

0
1
0
0
1
:
1
0
0

0
1
0
1
0
:
0
0
1

547B97DE172E6DB99F2B…
C427AE2AB58855F0F350…
906591E9352543748BAF…
F6EEDF5D3A594F2927F…
:

Minutia Triplets

Triangle Indexing

Triangle Hashing

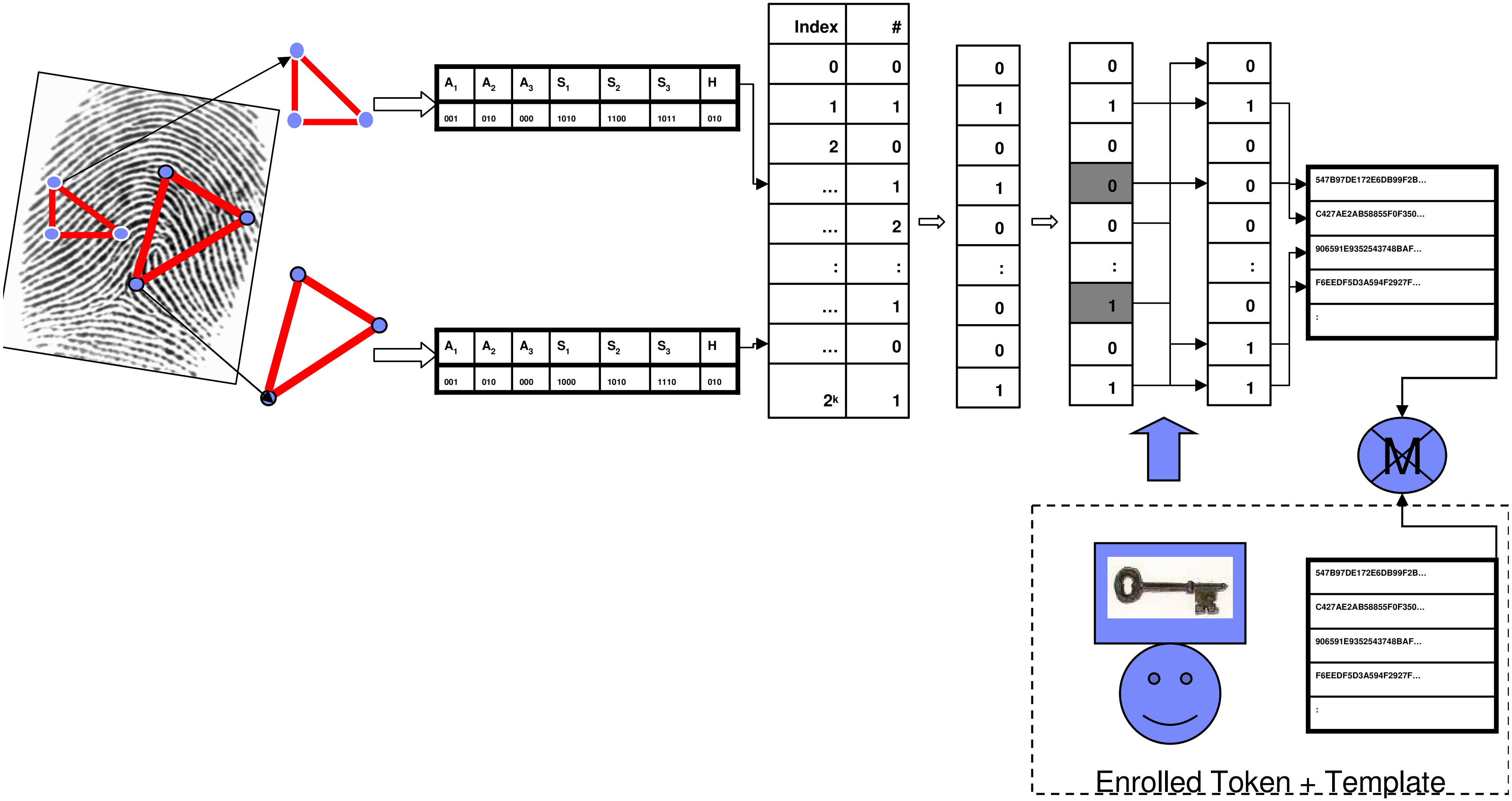
Binarization

Mutation

Randomization

Encryption

Verification



A1	A2	A3	S1	S2	S3	H
001	010	000	1010	1100	1011	010

A1	A2	A3	S1	S2	S3	H
001	010	000	1000	1010	1110	010

Index	#
0	0
1	1
2	0
…	1
…	2
:	:
…	1
…	0
2^k	1

0	0	0
1	1	1
0	0	0
1	0	0
0	0	0
:	:	:
0	1	0
0	0	1
1	1	1

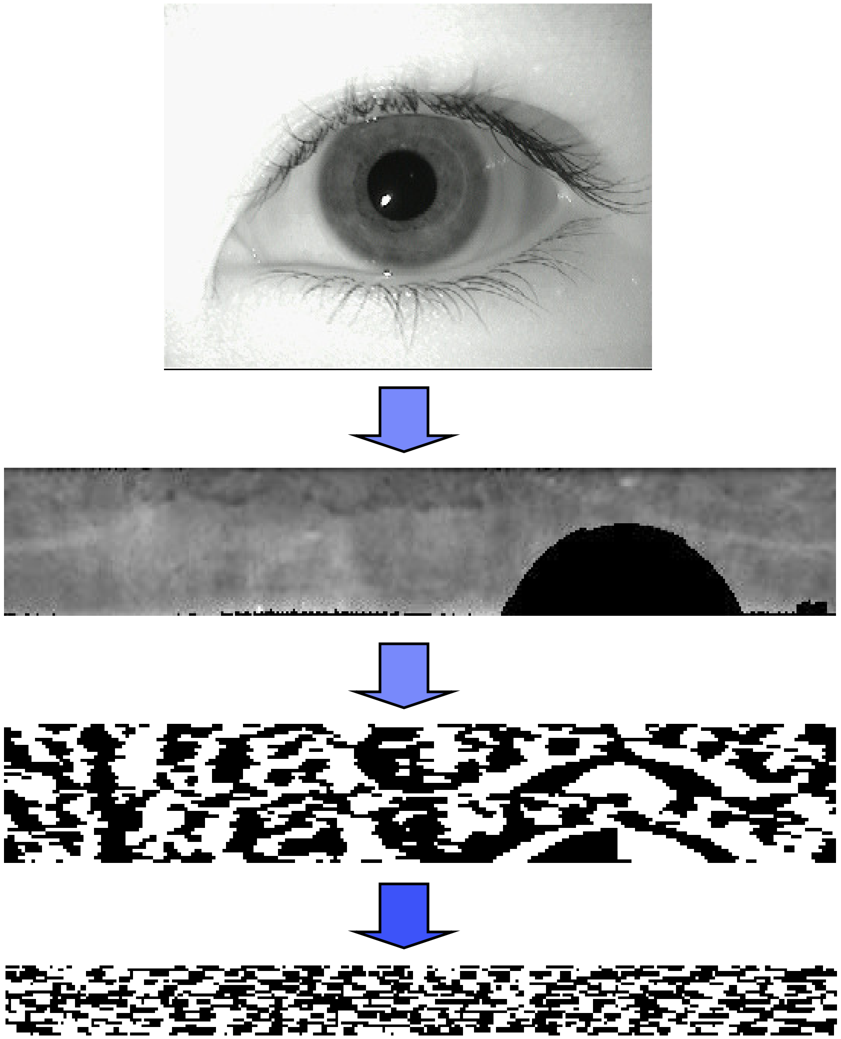
547B97DE172E6DB99F2B…
C427AE2AB58855F0F350…
906591E9352543748BAF…
F6EEDF5D3A594F2927F…
:

547B97DE172E6DB99F2B…
C427AE2AB58855F0F350…
906591E9352543748BAF…
F6EEDF5D3A594F2927F…
:

Enrolled Token + Template

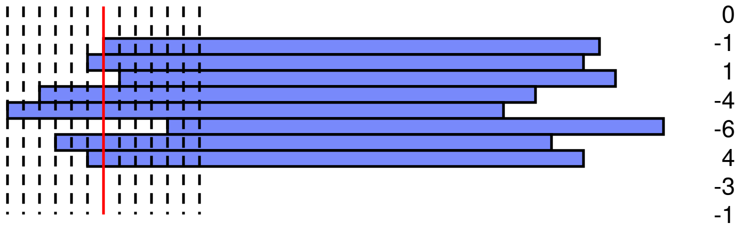
Steps in building a cancelable iris system

- **Segmentation**
- **Feature extraction**
- **Cancelable techniques ♦**



Method 1: GRAY COMBO

- **template based row shift and combination**
  - Step 1: for each row shift circularly:

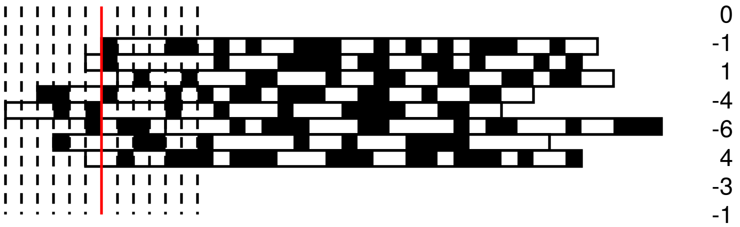


  - Step 2: combine two rows together to get a new one:
    - Intensity +, -
    - One row can be used more than once
    - Easy methods: odd+even, fold like a mirror

Combine rows 1, 3 to the new 1st row
Combine rows 2, 8 to the new 2nd row
Combine rows 4, 6 to the new 3rd row
Combine rows 5, 7 to the new 4th row

Method 2: BIN COMBO

- **code based row shift and combination**
  - Step 1: for each row shift circularly:

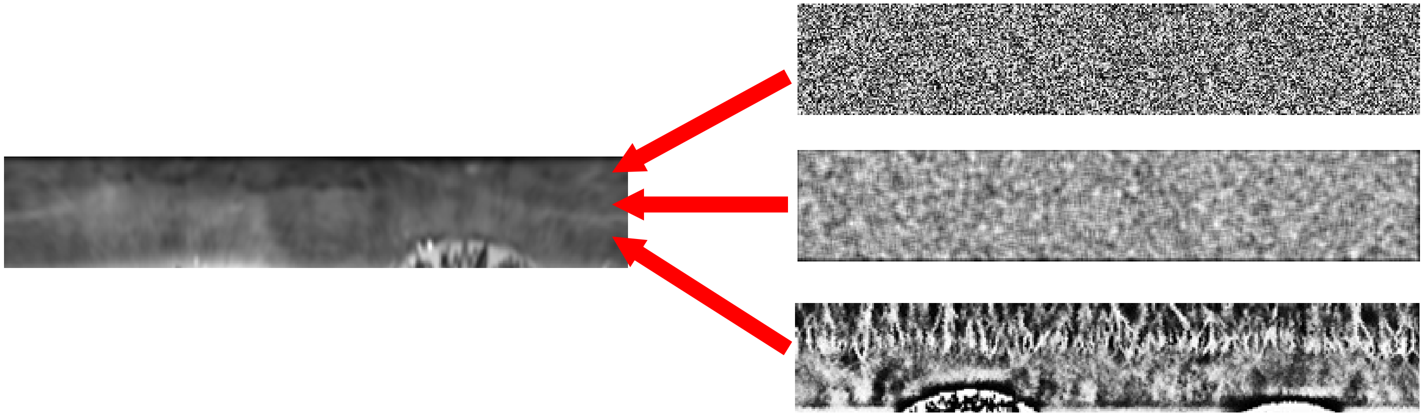


  - Step 2: combine two rows together to get a new one:
    - Binary XOR, or NXOR
    - One row can be used more than once
    - Easy methods: odd+even, fold like a mirror

Combine rows 1, 3 to the new 1st row
Combine rows 2, 8 to the new 2nd row
Combine rows 4, 6 to the new 3rd row
Combine rows 5, 7 to the new 4th row

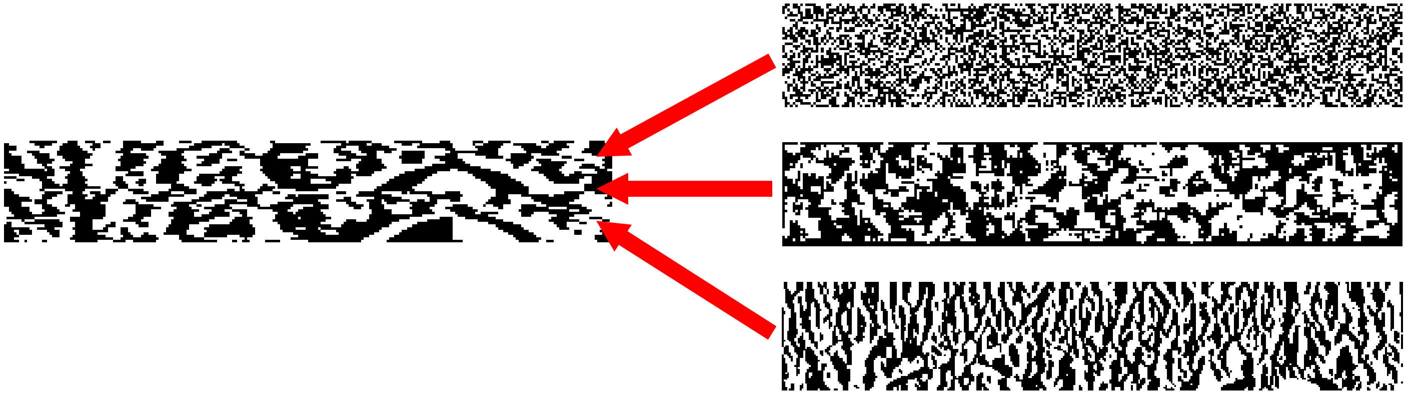
Method 3: GRAY SALT

- **template based salty noise**
  - Just plus a unique pattern --- random noise, random pattern or random synthetic iris texture
  - Generate new code according to the new texture



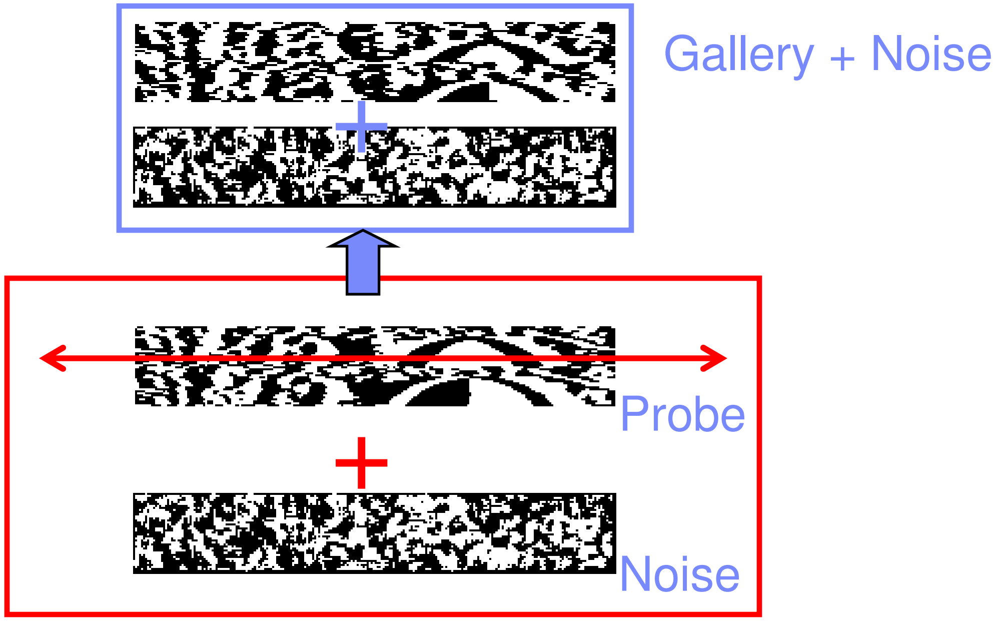
Method 4: BIN SALT

- **code based salty noise**
  - Just plus a unique binary pattern --- random noise , random pattern or random synthetic iris code



Matcher

- **Assume head tilt is not heavy**
- **Matching algorithm need to be modified:**



Key performance metrics

- **Accuracy**
  - How do the error rates change?
    - Same transform vs. different transform
- **Transform space**
  - How many transforms are possible?
  - Brute force non-invertible strength of the transform
- **Backward compatibility**
- **Impact on speed**

Thank you